Secure Coding in Go

Software Security and Risk Principles

Paul Mooney
Chief Software Architect, Microsoft MVP

@daishisystems | www.insidethecpu.com

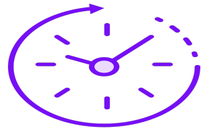
Coming Up

- Software Security Fundamentals
- OWASP and Go
- OWASP Top 10
- OWASP Top 10 Proactive Controls
- Mapping Security Requirements with the OWASP ASVS
- Demo: Forking and Customizing the ASVS GitHub Repository

The Fundamentals of Software Security

Confidentiality

Integrity



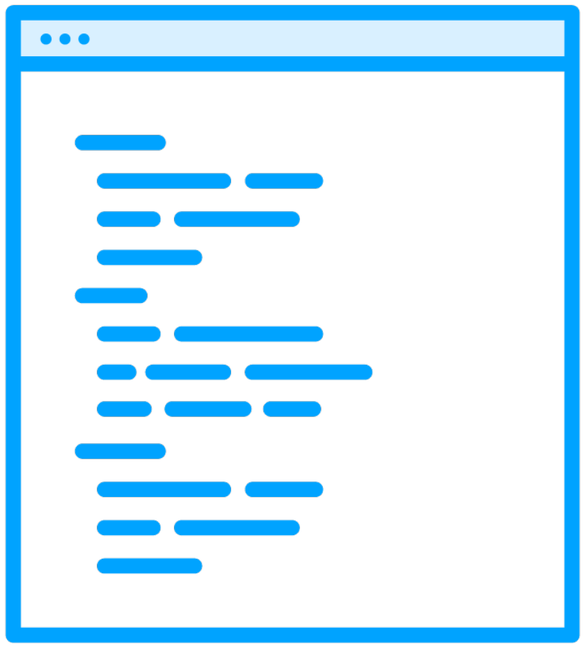
Availability

Authentication

Non-repudiation

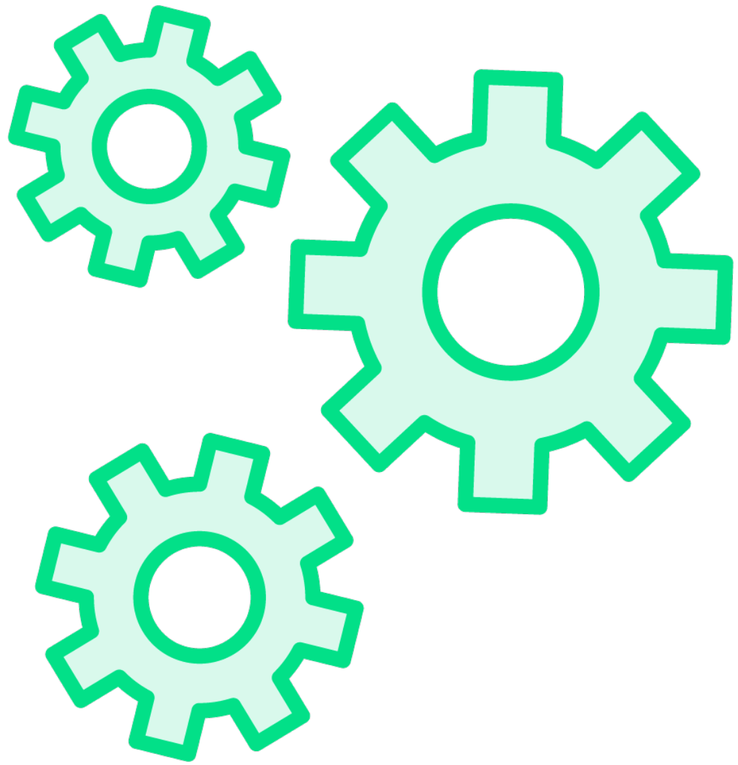
Incident response

Web Security and Go



- Statically typed
- Concurrent
- Garbage-collected

- Strict typing system
- Type confusion and coercion
- Concurrent programming support

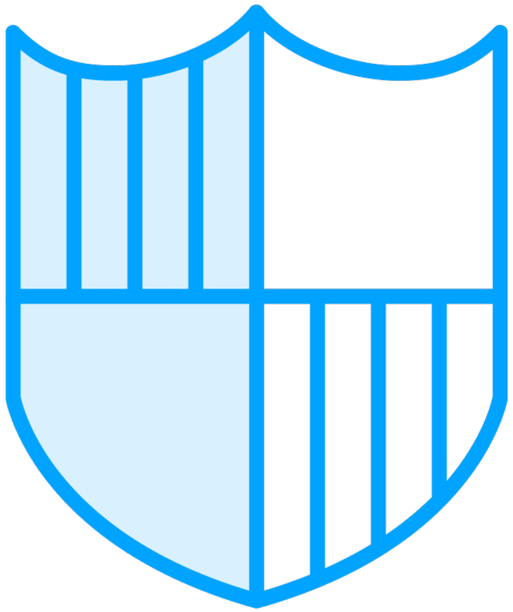


Built-in Security Control Libraries:

- Input validation
- Access control
- Cryptography
- Secure logging
- Auditing
- TLS

- Validate user input
- Secure authentication
- Session management

Open Web Application Security Project (OWASP)



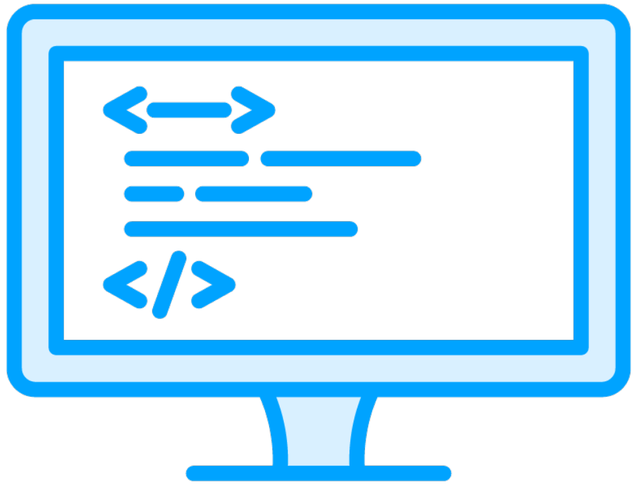
- Non-profit
- Improve web application security
- Freely available tools and resources

- OWASP Top 10
- 10 most critical security risks
- Updated regularly

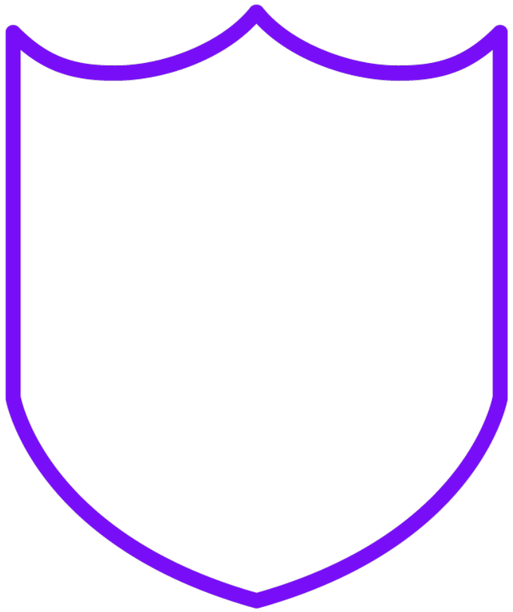
- Security terms glossary
- Secure coding guidelines
- Security vulnerability testing tools

OWASP and Go

- Simplicity
- Performance
- Security features

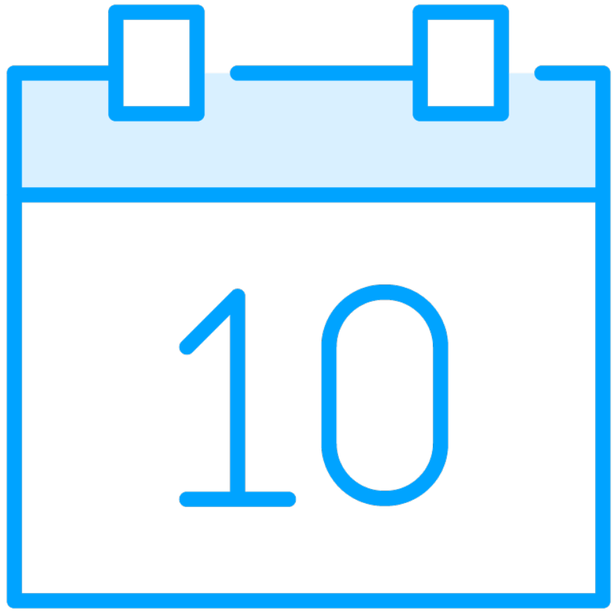


- Writing secure code
- Security packages and tools
- Code examples

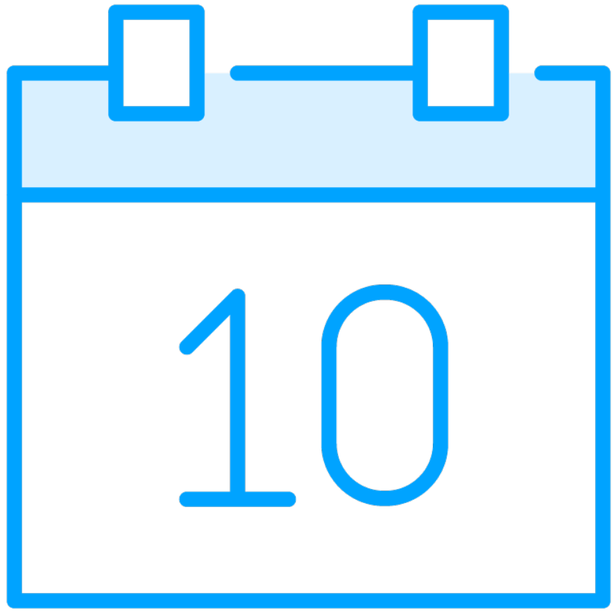


- OWASP Go Project
- Open source
- Community-driven
- Secure and stable framework
- Packages and libraries
- Guidance on writing secure code

The OWASP Top 10

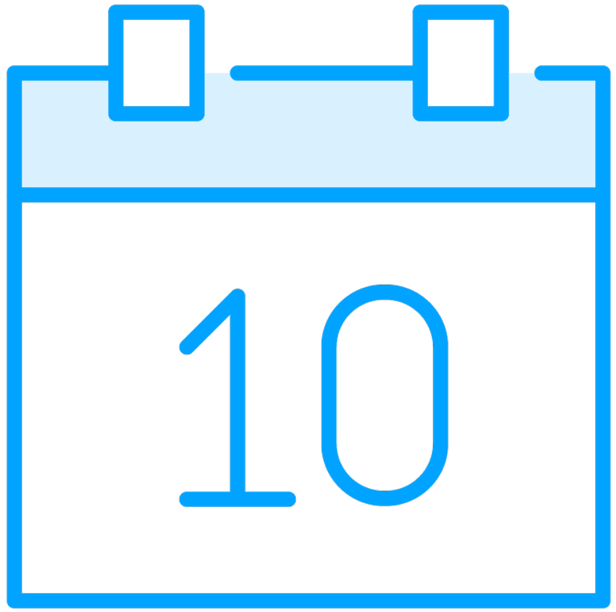


– 10 most critical application security risks
– Reflects current threat landscape
– Provides guidance for developers



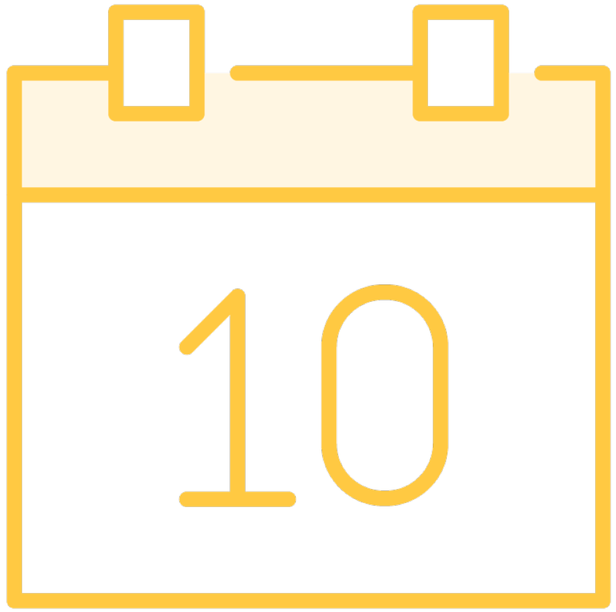
- Injection attacks
- Broken authentication
- Broken session management
- Broken access control
- Security misconfigurations
- Impact and ways to mitigate

Go and the OWASP Top 10



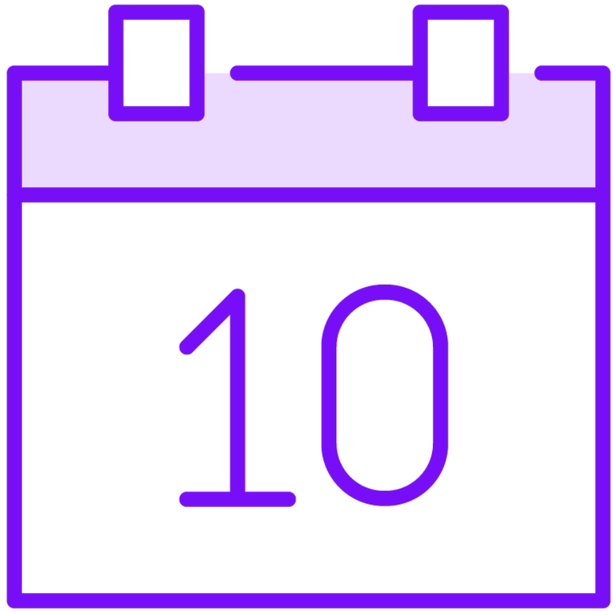
- Validate and sanitize user input
- SQL injection attacks
- Built-in libraries and functions
- Securely handle cookies and sessions
- Session hijacking
- Authentication and session management risks

The OWASP Top 10 Proactive Controls



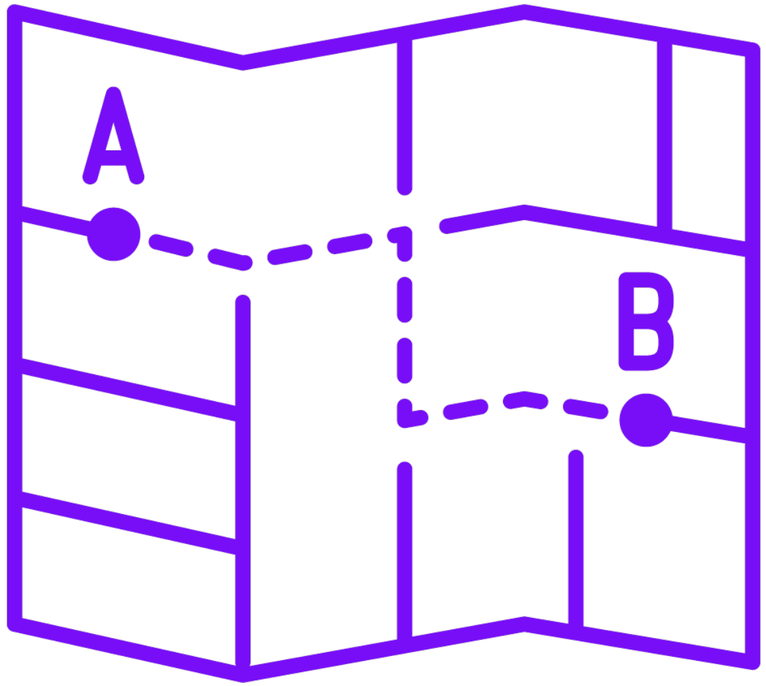
- Prioritized actionable recommendations
- Security program starting point
- Secure coding
- Access control
- Cryptography
- Incident response

Go and the OWASP Top 10 Proactive Controls



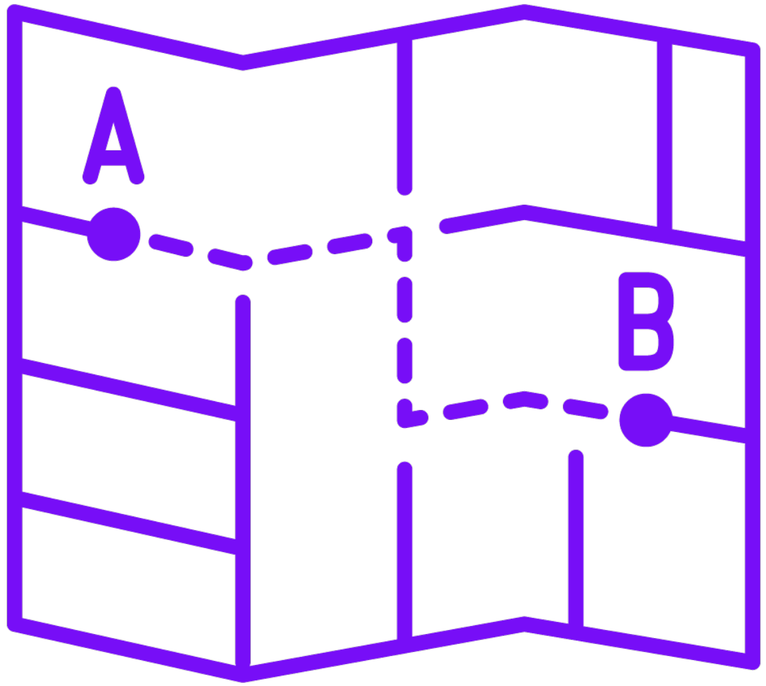
– Secure coding practices
– Pre-built security functions
– Open-source and community-driven

Mapping Security Requirements with the OWASP ASVS (Application Security Verification Standard)



– Verification requirements
– Functional requirements
– Technical requirements

- Level 1: Basic
- Level 2: Additional
- Level 3: Advanced



- Develop detailed security requirements
- Assess security during development
- Identify and prioritize areas for improvement
- Measure security maturity
- Provide a basis for test plans
- Compliance with security regulations

Demo

Forking and customizing the ASVS GitHub repository

Summary

- The Fundamentals of Software Security
- Web Security and Go
- OWASP and Go
- The OWASP Top 10
- Go and the OWASP Top 10 Proactive Controls
- Mapping Security Requirements with the OWASP ASVS (Application Security Verification Standard)